Sentence-level Planning for Especially Abstractive Summarization

Andreas Marfurt
Idiap Research Institute, Switzerland
EPFL, Switzerland
andreas.marfurt@idiap.ch

James Henderson
Idiap Research Institute, Switzerland
james.henderson@idiap.ch

Abstract

Abstractive summarization models heavily rely on copy mechanisms, such as the pointer network or attention, to achieve good performance, measured by textual overlap with reference summaries. As a result, the generated summaries stay close to the formulations in the source document. We propose the *sentence planner* model to generate more abstractive summaries. It includes a hierarchical decoder that first generates a representation for the next summary sentence, and then conditions the word generator on this representation. Our generated summaries are more abstractive and at the same time achieve high ROUGE scores when compared to human reference summaries. We verify the effectiveness of our design decisions with extensive evaluations.

1 Introduction

Abstractive summarization has improved drastically in recent years due to more efficient decoder architectures, like the Transformer (Vaswani et al., 2017), and language model pretraining, such as BERT (Devlin et al., 2019). As a result of these advances, current state-of-the-art models reach the performance of extractive systems, and even surpass them on some datasets (Liu and Lapata, 2019; Lewis et al., 2019; Zhang et al., 2020).

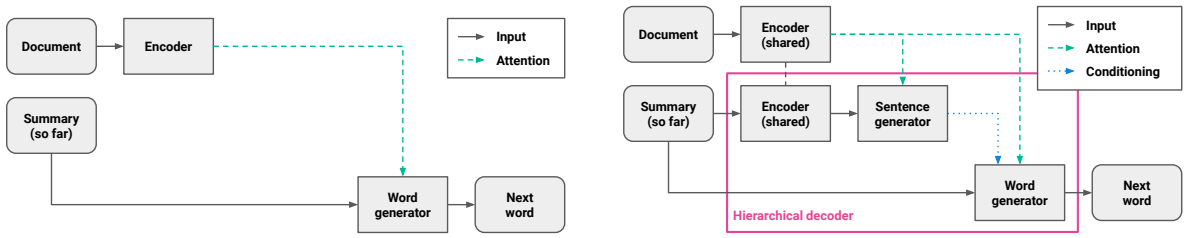
Part of this success, however, is due to the development of stronger copy mechanisms such as the pointer-generator network (See et al., 2017) or attention to the source document (Rush et al., 2015). The so-generated summaries copy long sequences from the input document, strung together with filler words. While this achieves better results in the predominant evaluation metric ROUGE (Lin, 2004), it comes at the cost of the summaries' abstractiveness and coherence, two qualities that we expect from human-written summaries.

In this paper, we aim to generate more abstractive summaries without sacrificing ROUGE and coherence. We achieve this by including a planning step at the sentence level before generating the summary word by word. The idea is to plan an outline for the next summary sentence first at a higher level to give the model more capacity for abstraction. As a result, the model has to rely less on copying the input, and thereby generates more abstractive summaries. Our model, the *sentence planner*, is an encoder-decoder architecture. The encoder is initialized from pretrained BERT weights. The decoder is hierarchical, and consists of a sentence generator that plans an outline for the summary at the sentence level, and a word generator that is conditioned on this outline when generating the summary's words. Both generators attend to the source document in order to condition their predictions on the input. The sentence planner is trained end-to-end to predict the words of the target summary, with an additional guidance loss that encourages the sentence generator to produce the encoder's embedding for the target next sentence. This is the first work to propose a hierarchical Transformer decoder that generates a summary from latent sentence representations.[1]

We extensively evaluate our model on a recently published highly abstractive dataset and an established but more extractive corpus. We show that the sentence planner generates more abstractive summaries while improving the ROUGE scores of a state-of-the-art model without a hierarchical decoder. We use gradient attribution to quantify the impact of the sentence generator on the model's prediction as well as how much information from the document it captures. Moreover, we verify the effectiveness of our model components with an ablation study, and show that simply increasing the baseline's decoder parameters does not bring it up

[1] Our code is available at https://github.com/idiap/sentence-planner.

Proceedings of the Third Workshop on New Frontiers in Summarization, pages 1–14
November 10, 2021.



(a) BERTSUMEXTABS (b) Sentence planner

Figure 1: (a) BERTSUMEXTABS model. An encoder encodes the document, and a word generator generates the next word given previous words, while paying attention to the document. (b) Sentence planner model. A shared encoder separately encodes the document and each sentence of the summary generated so far. The sentence generator takes the summary sentence embeddings and predicts the next sentence embedding, which the word generator is then conditioned on. Both generators integrate document information through attention.

to par with the hierarchical decoder. Our automatic evaluations are confirmed in a human evaluation study, where the sentence planner improves upon its strong baseline in each of six quality categories.

Our contributions are twofold: (a) We are the first to propose a hierarchical Transformer decoder that generates summaries from a latent sentence-level plan, and (b) we perform an extensive evaluation of our model on two summarization datasets and show that it produces more abstractive summaries while retaining high ROUGE scores, two objectives that are in opposition.

2 The Hierarchical Decoder

Our approach builds on the BERTSUMEXTABS model (Liu and Lapata, 2019). Their model consists of an encoder initialized with an extractive summarization model, which in turn was initialized with a BERT model, and a randomly initialized Transformer decoder.[2] We keep the encoder the same. We replace the decoder with a hierarchical version by introducing a sentence generator that develops a high-level plan for the summary, and a word generator that is conditioned on this plan. A model diagram is shown in Figure 1. Section 2.1 describes how the sentence generator develops the outline for the summary, and Section 2.2 shows how the word generator makes use of it.

2.1 Sentence Generator

The sentence generator is a two-layer Transformer decoder. It receives as inputs the sentence representations of completed summary sentences, and generates a sentence representation for the next summary sentence.

Inputs. The inputs to the sentence generator are a sequence of representations of already completed summary sentences. These are computed by the same encoder that computes representations for the document tokens. For each individual previous summary sentence, the encoder computes its contextualized token embeddings. We use the contextual embedding of the end-of-sentence token as a representation for the sentence.[3] When generating the first summary sentence, there are no completed sentences, so we use a single zero vector as input to the sentence generator.

During training with teacher forcing, we use the previous portion of the reference summary as input to the encoder. Since the entire summary is known in advance, we can compute all inputs to the sentence generator in parallel.

Self-attention. The sentence generator's self-attention operates at the sentence level, which means the sequence length n for our Transformer decoder is very small (between 2 and 4 on average, see Section 4). As a result, the self-attention computation, which is quadratic in the sequence length, becomes extremely cheap. As in regular Transformer decoders, a causal mask prevents attention to future sentences.

Cross-attention. In the cross-attention, the sentence generator pays attention to the encoded document. Through this connection, the sentence generator is able to compare the already generated

[2] Even stronger results have recently been achieved when pretraining an entire sequence-to-sequence model on a task closer to summarization (BART (Lewis et al., 2019), PEGASUS (Zhang et al., 2020)). In this paper, we restrict ourselves to encoder initializations with the BERT model and do not consider other pretraining approaches, since these techniques are orthogonal to our contribution.

[3] We found that this performed better than alternative encodings of the summary, as discussed in Appendix A.

Dataset	Examples	Mean doc length		Mean summary length		Novel bigrams	Corefs
		words	sentences	words	sentences		
CNN/DailyMail	312085	685.12	30.71	52.00	3.88	54.33%	0.105
Curation Corpus	39911	504.26	18.27	82.63	3.46	69.22%	0.441

Table 1: Dataset statistics.

summary to the document and identify missing information that should appear in the next sentence.

Output. The output of the sentence generator is a representation r_{sent} for the next summary sentence. Section 2.2 describes how we condition the word generator on this sentence representation.

Guidance loss. We provide the sentence generator with an additional loss term for guidance. Since during training, we know the ground truth next summary sentence and can compute its encoding r_{gold}, we penalize the (element-wise) mean squared error between the gold and the predicted next sentence representation.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{d}\sum_{i=1}^{d} ||r_{\text{gold}}^{(i)} - r_{\text{sent}}^{(i)}||_2^2 \quad (1)$$

where d is the representations' dimension. This loss term is added to the regular cross-entropy loss with a scaling hyperparameter λ, although we found $\lambda = 1$ to work well in practice.

We do not backpropagate the guidance loss's gradients from the sentence generator into the encoder to avoid a collapse to a trivial solution. Otherwise, the encoder might output the same representation for every sentence so that the sentence generator can perfectly predict it.

2.2 Word Generator

Our word generator is also a Transformer decoder. The regular Transformer decoder consists of layers l with self-attention, cross-attention and feed-forward sublayers. They are defined as follows:

$$s^l = \text{LN}(h^{l-1} + \text{SelfAtt}(h^{l-1})) \quad (2)$$

$$c^l = \text{LN}(s^l + \text{CrossAtt}(s^l, r_{\text{enc}})) \quad (3)$$

$$h^l = \text{LN}(c^l + \text{FFN}(c^l)) \quad (4)$$

where LN is layer normalization (Ba et al., 2016), SelfAtt stands for self-attention, CrossAtt is the cross-attention to the encoder outputs r_{enc}, and FFN is the feed-forward sublayer consisting of two fully-connected layers with an intermediate non-linearity.

In our word generator, we condition on the sentence representation by replacing Eq. 3 with

$$c^l = \text{LN}(s^l + \text{CrossAtt}(s^l, r_{\text{enc}}) + r'_{\text{sent}}) \quad (5)$$

where r'_{sent} is the sentence representation obtained from the sentence generator, passed through a fully-connected and a dropout layer. We do not differentiate between layers and add the same sentence representation in every layer and to every token.

We experimented with various ways to use attention in the word generator to integrate the sentence representation. However, the conditioning method presented above substantially outperforms the attention-based integrations of the sentence representation. We further discuss this topic in Appendix A.

At the end of a sentence, the word generator either outputs a special sentence separator symbol, prompting the sentence generator to generate the next sentence representation, or an end-of-summary symbol, stopping generation.

3 Experimental Setup

We now describe the datasets (§ 3.1) and metrics (§ 3.2) that we use to evaluate our model, and give implementation details (§ 3.3) to replicate our experiments. Dataset statistics are shown in Table 1.

3.1 Datasets

CNN/DailyMail. The CNN/DailyMail corpus was initially introduced as a question answering dataset in Hermann et al. (2015) and adapted for summarization by Nallapati et al. (2016), and has been widely used. The corpus's summaries are a concatenation of bullet points describing the highlights of the news article. They are therefore designed to be concise, but do not necessarily form a fluent summary. Extractive approaches perform well on CNN/DailyMail (Liu and Lapata, 2019).

Curation Corpus. The Curation Corpus (Curation, 2020) is a recently introduced dataset of professionally written summaries of news articles. The corpus is an order of magnitude smaller than

CNN/DailyMail, and its articles and summaries have fewer but longer sentences (see Table 1). We see this dataset as better representing the summarization task, since the summaries were written for this purpose specifically. Additionally, Curation Corpus's summaries span multiple sentences, in contrast to a dataset such as XSum (Narayan et al., 2018), which is a prerequisite for our approach. As a consequence, the majority of our experiments are conducted on Curation Corpus (see Section 4). We describe our preprocessing in Appendix B.

3.2 Metrics

ROUGE. The standard metric to automatically evaluate summarization systems is the ROUGE F1 score (Lin, 2004). It measures textual overlap between the generated candidate and the reference summaries. The length of text spans for computing the overlap can be arbitrary, but it is common to report unigram and bigram overlap (ROUGE-1, ROUGE-2), as well as the longest common subsequence (ROUGE-L).

Novel bigrams. The fraction of novel bigrams in the generated summary with respect to the source document measures its abstractiveness. More abstractive methods generally attain lower ROUGE scores. To see why, consider the case where the reference summary and the model copy from the document. The generated summary is guaranteed to get an exact match and high ROUGE. In the opposite case, where both the reference summary and the model generate novel text, there is a good chance that the choice of words is not exactly the same, resulting in low ROUGE.

Corefs. Inspired by Iida and Tokunaga (2012), we evaluate discourse coherence with a coreference resolution model. We count the number of coreference links across sentence boundaries as a proxy for the coherence of a summary, i.e. whether the sentences build upon information in the preceeding ones. Since summaries with more sentences could be favored by this count, we normalize by the number of sentences. To extract coreferences from the generated summaries, we use the neuralcoref[4] implementation. Table 1 shows the mean number of coreference links across sentence boundaries for the datasets' reference summaries. We clearly see that the summaries in the Curation Corpus are written in a much more coherent style than the ones from CNN/DailyMail. Specifically, the bullet point style summaries in CNN/DailyMail do not foster summaries whose sentences build on each other. However, this is a quality we would expect from human summaries, which is yet another reason to focus our analysis on the Curation Corpus.

3.3 Implementation Details

We use the code from BERTSUMEXTABS[5] for our experiments. For the decoder, they have their own Transformer implementation while we employ the popular huggingface library (Wolf et al., 2019). In our experiments, we control for the possible discrepancy between these two implementations by reporting BERTSUMEXTABS's performance with a huggingface Transformer as well.

We use the hyperparameters from BERTSUMEXTABS where not specified otherwise. For our implementation, a grid search found a learning rate of 0.001 for the BERT-initialized encoder and 0.02 for the randomly initialized Transformer(s) to work best. We use a fixed batch size of 3 with gradient accumulation over 5 batches. The hyperparameters for our implementation of BERTSUMEXTABS and our model are exactly the same, and we only tune the hyperparameters of the sentence generator with a grid search.

Our sentence generator is a 2-layer Transformer with 12 heads, a hidden size of 768, an intermediate dimension of 3072 for the feed-forward sublayer, and dropout of 0.1 for attention outputs. We do not apply dropout to the outputs of linear layers.

Curation Corpus. All our models are trained for 40,000 training steps, with a learning rate warmup of 2,500 steps. We did not see an improvement from initializing the encoder with a pretrained extractive model, and therefore initialize from BERT weights. We average the results from 5 runs, and also report the standard deviation in Appendix C.

CNN/DailyMail. Our models are trained for 200,000 training steps, with 20,000 warmup steps for the pretrained encoder, and 10,000 warmup steps for the randomly initialized Transformer(s), following Liu and Lapata (2019). We also use their model checkpoint of BERTSUMEXT to initialize the encoder in all our models.

[4] https://github.com/huggingface/neuralcoref

[5] https://github.com/nlpyang/PreSumm

Model	ROUGE			Sentences		Novel Bigrams	Corefs
	R-1	R-2	R-L	Number	Length		
Gold summaries	-	-	-	*3.46*	*28.0*	*69.22%*	*0.441*
BSEA (Liu and Lapata, 2019)	42.95	17.67	37.46	2.73	27.3	36.77%	0.267
BSEA (our implementation)	43.37	17.92	37.73	2.76	28.5	37.29%	0.283
Sentence planner	**44.40**	**18.31**	**38.69**	3.15	28.2	**39.29%**	**0.289**

Table 2: Results on Curation Corpus. Mean over 5 runs. Best result in bold.

Model	IG	Conductance
BSEA	-	-
+ Sentence generator	25.1%	32.3%
+ $\mathcal{L}_{\text{MSE}}$ (= Sentence planner)	36.6%	29.1%

Table 3: Attribution study. IG: Attribution of the model predictions to r_{sent} vs. to cross-attention. Conductance: Attribution of the predictions to the article via r_{sent} vs. via cross-attention.

4 Results

We now turn to evaluation of our method. First, we show the results on Curation Corpus (§ 4.1). With attribution techniques (§ 4.2) and an ablation study (§ 4.3) we uncover how the model uses the sentence generator component. Increasing the number of parameters of BERTSUMEXTABS (BSEA) does not provide the same improvements as our approach (§ 4.4). On the CNN/DailyMail dataset, our model generates more abstractive summaries while retaining high ROUGE scores (§ 4.5). Finally, a human evaluation validates the results from our automatic metrics (§ 4.6).

4.1 Results on Curation Corpus

Table 2 shows the results of our evaluation on the Curation Corpus. The sentence planner substantially improves ROUGE scores compared to BERTSUMEXTABS. The relative difference is between 2.2% and 2.5% for the different ROUGE variants. A noticeable difference also exists between the ROUGE scores of the two base model implementations, which is why we continue reporting the scores for both in the following.

The sentence planner's summaries are more abstractive than those of BERTSUMEXTABS, as indicated by the number of novel bigrams. However, there is still a large gap to the reference summaries displayed on the first line. The sentence planner generates substantially more sentences than BERTSUMEXTABS on average, moving it closer to the gold summaries. The mean number of words within those sentences stays close to the reference statistic. [6]

The mean number of coreferences across sentence boundaries, normalized by the number of sentences, is similar for all models, with the best score achieved by the sentence planner. This number is lower than for the reference summaries but substantially higher than for references and generated summaries from the CNN/DailyMail corpus (see Section 4.5).

4.2 Attribution to Sentence Representation

A natural question to ask is whether the sentence representation r_{sent} is actually used by the word generator. We therefore compare the attribution of the model predictions to r_{sent} with the attribution to the output of the cross-attention. We use the Integrated Gradients (IG) algorithm (Sundararajan et al., 2017) with respect to these intermediate representations. We choose the zero vector as a baseline r_0, but taking the mean of r_{sent} over the test examples as a baseline provides similar results. We then integrate along the path from r_0 to r_{sent}

$$(r_{\text{sent}} - r_0) \int_{\eta=0}^{1} \frac{\partial F(x, r_0 + \eta(r_{\text{sent}} - r_0))}{\partial r_{\text{sent}}} \quad (6)$$

for a given input x. In practice, we discretize the integral and sum over 50 integration steps with linearly spaced η values. The case for the attribution to the cross-attention output is analogous. We report the relative attribution to r_{sent} in Table 3. The result is averaged over the first 100 examples in our

[6] The mean number of sentences and (to a lesser extent) their average length can be influenced by a length penalty hyperparameter α, which is set between 0.6 and 1 (Liu and Lapata, 2019). BERTSUMEXTABS with no penalty ($\alpha = 1$) produces the same number of sentences and words as the sentence planner with the largest penalty ($\alpha = 0.6$), but a large gap in ROUGE-(1/2/L) remains: $(0.7/0.6/0.6)$. Consistent with Sun et al. (2019), we find that ROUGE scores increase with length and α, but we also find that novel bigrams decrease. In order to not favor one side of the trade-off over the other, we stick with the setting of $\alpha = 0.95$ from Liu and Lapata (2019) for both models.

Model	ROUGE		
	R-1	R-2	R-L
BSEA (our implementation)	43.37 (0.37)	17.92 (0.17)	37.73 (0.31)
+ Sentence generator	43.97 (0.30)	18.28 (0.11)	38.32 (0.22)
+ $\mathcal{L}_{\text{MSE}}$ (= Sentence planner)	**44.40** (0.14)	**18.31** (0.13)	**38.69** (0.10)

Table 4: Ablation study showing ROUGE scores on Curation Corpus when adding the individual components of our model. Mean and std (in brackets) over 5 runs.

Model	Parameters	ROUGE			Novel Bigrams
		R-1	R-2	R-L	
BSEA (Liu and Lapata, 2019, $L_{\text{dec}} = 6$, $\text{ff}_{\text{dec}} = 2048$)	180M	43.13	17.80	37.63	36.83%
BSEA (our implementation, $L_{\text{dec}} = 6$, $\text{ff}_{\text{dec}} = 2048$)	182M	43.21	17.69	37.54	37.12%
BSEA (our implementation, $L_{\text{dec}} = 6$, $\text{ff}_{\text{dec}} = 3072$)	191M	43.12	17.84	37.53	37.34%
BSEA (our implementation, $L_{\text{dec}} = 8$, $\text{ff}_{\text{dec}} = 2048$)	198M	43.41	17.91	37.79	37.09%
BSEA (our implementation, $L_{\text{dec}} = 8$, $\text{ff}_{\text{dec}} = 3072$)	210M	43.68	18.06	38.06	37.77%
Sentence planner	208M	**44.40**	**18.31**	**38.69**	**39.29%**

Table 5: Number of parameters of each model (M = million) together with ROUGE scores and novel bigrams on Curation Corpus.

test set. It shows that the attribution to r_{sent} with the sentence generator alone is about a quarter, while three quarters are attributed to the cross-attention. This is already a substantial amount, considering that the alternative is to directly look at the document. r_{sent}'s attribution share further increases to more than a third with the addition of the guidance loss $\mathcal{L}_{\text{MSE}}$, making r_{sent} even more useful.

While we expect that the sentence representation is mostly used as an outline for the next summary sentence, we are curious to see how much information of the source document is present in r_{sent}. We use the conductance (Dhamdhere et al., 2019) via r_{sent} with respect to the encoder outputs, and compare it to the conductance via the cross-attention. We ignore the encoder's computation as it is the same for both paths. Since it is computationally expensive to compute gradients over every neuron in r_{sent}, we sum over just 5 integration steps and average the result over the first 10 examples of the test set. From Table 3, we see that almost a third of the document's information is passed through the sentence representation. The addition of the guidance loss decreases this number, which means that r_{sent} serves more as an outline than an additional condensed representation of the document.

4.3 Model Ablation

Table 4 shows an ablation study for the two components we introduced in the hierarchical decoder. Both the sentence generator network and the guidance loss provide a steady increase in ROUGE performance as well as a reduction in variance. This demonstrates the efficacy of our additions.

4.4 Number of Parameters

To verify that the improved performance of the sentence planner is not just a result of the increased number of parameters, we perform an experiment where we increase the base model's capacity. BERTSUMEXTABS consists of a 12-layer Transformer encoder, and a 6-layer decoder. Our model has additional parameters in the 2-layer Transformer that serves as the sentence generator. We therefore increase the BERTSUMEXTABS decoder's parameters such that the total model sizes match. Specifically, we increase the number of layers L_{dec} and the inner dimension of the feed-forward sublayer ff_{dec}. The comparison is shown in Table 5. While increasing the number of parameters improves BERTSUMEXTABS's ROUGE scores, they are still far behind the sentence planner's scores. Similarly, the share of novel bigrams rises a bit with additional parameters. However, it still stays behind the abstractiveness of the sentence planner, showing that the inductive bias of our hierarchical decoder is very effective.

4.5 Results on CNN/DailyMail

For comparison with previous work, we now report the results on the more extractive CNN/DailyMail corpus. Table 6 shows the results for BERTSUMEXTABS and the sentence planner. The first

Model	ROUGE			Sentences		Novel Bigrams	Corefs
	R-1	R-2	R-L	Num	Len		
Gold summaries	-	-	-	*3.88*	*14.08*	*54.33%*	*0.105*
BSEA (Liu and Lapata, 2019, their checkpoint)	42.16	19.49	39.16	3.33	19.1	7.40%	0.124
BSEA (Liu and Lapata, 2019, our training)	41.17	18.82	38.27	3.07	18.5	8.14%	0.126
BSEA (our implementation)	41.48	18.86	38.41	2.99	19.6	7.18%	0.104
Sentence planner	41.87	19.37	39.02	3.82	17.8	**10.65%**	**0.132**

Table 6: Results on CNN/DailyMail. Best result with our own training underlined.

Quality	BSEA	SP	p-value
Non-redundancy	4.05	**4.08**	0.408
Fluency	3.70	**3.75**	0.343
Structure/coherence	3.68	**3.85**	0.102
Informativeness	3.57	**3.77**	0.069
Abstractiveness	3.45	**3.65**	0.047
Semantic similarity	2.98	**3.18**	0.043

Table 7: Mean score for each quality in the human evaluation for BSEA and the sentence planner (SP). Scores range from 1 (worst) to 5 (best). The p-value is determined with a paired bootstrap test.

line evaluates the model checkpoint that Liu and Lapata (2019) provide. When we train both the extractive initialization and the abstractive model ourselves with the hyperparameters suggested, we are not quite able to achieve the same results. With our own implementation of the decoder, we are able to close the gap in ROUGE scores somewhat. The sentence planner performs best out of the models we trained ourselves. As on the Curation Corpus, it is also much more abstractive than BERTSUMEXTABS. This could well account for the remaining difference in ROUGE scores.

The mean number of generated sentences by the sentence planner is almost identical with the reference summaries, and again a lot larger than for BERTSUMEXTABS. The generated sentences are also shorter, in line with the references. The number of coreference links across sentence boundaries are similar across models, with the sentence planner producing those links most often. We conclude that even on the more extractive CNN/DailyMail corpus, the sentence planner generates more abstractive and coherent summaries at high ROUGE.

4.6 Human Evaluation

We perform a human evaluation to verify the results found by our automatic metrics. We compare outputs of BERTSUMEXTABS (our implementation) with the sentence planner. The annotators are presented with the source article, the reference summary as well as the candidate summaries for both systems. The systems are labeled *1* and *2*, and their order is randomized for each example. For each candidate summary, the annotators then have to select a score from 1 to 5 for six qualities, which are presented with a descriptive question (in brackets). The qualities are non-redundancy (*Is information stated only once?*), fluency (*Is the summary grammatical and good to read?*), structure/coherence (*Do the sentences build on each other?*), informativeness (*Is the important information captured?*), abstractiveness (*How much of the summary is rephrased (instead of copied)?*), and semantic similarity (*How semantically similar is the candidate summary to the gold summary?*).

We randomly draw 20 examples from the Curation Corpus test set. We limit the number of words of source articles to be above 100 and below 700 (includes 70% of examples), to remove extreme examples and keep the workload for annotators reasonable. We divide our 6 annotators, which are all NLP experts, into two groups, who review 10 examples each, resulting in 3 annotations per example, of which we take the mean. The results are reported in Table 7. The sentence planner is evaluated favorably compared to BERTSUMEXTABS in all categories. The non-redundancy and fluency categories show a smaller gap. This is expected, as we did not change the word generator, which impacts these categories the most. On the other categories, the sentence planner achieves larger improvements, showing that the introduction of a hierarchical decoder improves the planning capabilities of the model.

To determine statistical significance of the results, we follow the guidelines in Dror et al. (2018) and select the non-parametric paired bootstrap test (Efron and Tibshirani, 1994). We find that the two models are not significantly different for the first four categories, while they are for the abstrac-

tiveness and semantic similarity categories when selecting a threshold of $p = 0.05$. Additionally, we quantify the inter-annotator reliability with the intraclass correlation coefficient (ICC), according to Shrout and Fleiss (1979). The reliability is moderate with an ICC of 0.56 and a 95% confidence interval of [0.46, 0.65]. Given the moderate annotator agreement and our relatively small sample size of the human evaluation, it is possible that a more extensive (and therefore expensive) human evaluation could show a significant difference in informativeness and structure/coherence.

Finally, we are curious whether the Corefs evaluation can serve as an automatic evaluation of the structure/coherence category. We therefore compute the Pearson ρ for the correlation between the human and the metric's scores. The correlation is weak at 0.098 (p-value: 0.549). Thus there seems to be a mismatch between what the metric measures (discourse coherence by counting the number of coreference links across sentence boundaries) and the open way the question was formulated in the human evaluation (*Do the sentences build on each other?*). Nevertheless, the Corefs metric showed its value by very clearly distinguishing the CNN/DailyMail's summaries from the Curation Corpus's summaries. We therefore leave its optimal use for future work.

5 Related Work

Our proposal is the first hierarchical decoder for a Transformer-based text summarization model. We survey previous work on hierarchical attention in summarization on the encoder side, and sentence-level planning on the decoder side.

5.1 Hierarchical Attention

Nallapati et al. (2016) use hierarchical attention in the encoder with a word- and a sentence-level RNN. The attention weights at the word level are re-weighted by the sentence-level attention weights. Celikyilmaz et al. (2018) divide the document into paragraphs, which are encoded separately by agents. Each agent performs attention within its paragraph, and the decoder attends to the agents. Gehrmann et al. (2018) first employ a content selector at the word level to decide which words are candidates for copying. They then use a pointer-generator network with just the admissible tokens to generate the summary. Miculicich et al. (2018) use hierarchical attention networks (Yang et al., 2016) to encode the context of previous sentences, which is used to inform the translation of the next word. In contrast to these methods, we employ hierarchy on the decoder side, and generate a sentence representation for the *next* sentence.

5.2 Sentence Planning

Tan et al. (2017) use word- and sentence-level RNNs in both encoder and decoder. They also predict a next sentence embedding, but use a graph model as importance for the encoded sentences instead of attention. The word-level decoder RNN is conditioned by initializing the first hidden state with the sentence embedding. Perez-Beltrachini et al. (2019) use a CNN word encoder/decoder and an LSTM sentence decoder for multi-document summarization. They predict a next sentence embedding with attention, which they add to the input of each convolutional decoder layer. An auxiliary loss pushes sentence embeddings to be close to LDA topics of summary sentences. Both models do not employ Transformers, and consequently their conditioning is very different from ours.

Several papers have investigated sentence-level language modeling. Ippolito et al. (2020) pick the most likely continuation from a set of candidate sentences. Their task provides a context of four sentences and requires to pick a single following sentence. A pretrained BERT model generates a target sentence representation, and the candidate with the highest cosine similarity is selected. Huang et al. (2020) address the task of sentence infilling, where context on both sides of the missing sentence is provided. They learn sentence representations with a denoising autoencoder, predict the representation of the missing sentence with a separate Transformer, and then use the autoencoder's decoder to generate the missing sentence from that representation. Deutsch and Roth (2019) propose the *summary cloze* task. Given the beginning of a summary, the topic and the reference document, their model has to continue with a single sentence supported by the reference document. These approaches only predict a single sentence, and are given substantial context. In our approach, we generate sentence representations with variable context (or no context for the first summary sentence).

Hua and Wang (2020) receive a prompt and a set of keyphrases, which they position and then fill in the gaps around them. Similarly, Jhamtani and Berg-Kirkpatrick (2020) generate a keyword per

sentence to be generated, and then generate its left and right context. In contrast to these approaches, our sentence generator outputs a latent representation r_{sent} for the entire sentence, which is used to condition the word generator. We do not tie this representation to specific words.

6 Conclusion

We presented the sentence planner, an encoder-decoder model with a hierarchical decoder, consisting of a sentence and a word generator. Our sentence generator computes a plan for the next summary sentence. The word generator is then conditioned on this plan when generating the sentence's words. An additional loss term, which guides the sentence planner towards producing the embedding of the target next sentence, improves the sentence generator's plan. When comparing the sentence planner to a state-of-the-art model without a hierarchical decoder, it generates more abstractive and coherent summaries at higher ROUGE scores.

In future work, we aim to apply our model to other generation tasks, such as machine translation or dialogue generation.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675.

Curation. 2020. Curation corpus base.

Daniel Deutsch and Dan Roth. 2019. Summary cloze: A new task for content selection in topic-focused summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3711–3720.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. 2019. How important is a neuron. In *International Conference on Learning Representations*.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Xinyu Hua and Lu Wang. 2020. PAIR: Planning and iterative refinement in pre-trained transformers for long text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yichen Huang, Yizhe Zhang, Oussama Elachqar, and Yu Cheng. 2020. INSET: Sentence infilling with inter-sentential transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2502–2515.

Ryu Iida and Takenobu Tokunaga. 2012. A metric for evaluating discourse coherence based on coreference resolution. In *24th International Conference on Computational Linguistics*.

Daphne Ippolito, David Grangier, Douglas Eck, and Chris Callison-Burch. 2020. Toward better storylines with sentence-level language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7472–7478.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2020. Narrative text generation with a latent discrete plan. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Patrick E Shrout and Joseph L Fleiss. 1979. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2):420.

Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Stratos Xenouleas, Prodromos Malakasiotis, Marianna Apidianaki, and Ion Androutsopoulos. 2019. Sumqe: A bert-based summary quality estimation model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6007–6013.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Jingqing Zhang, Yao Zhao, Mohammed Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*.

A Alternative Approaches

In the following, we discuss alternative approaches which we tried but did not achieve as good results as the proposed model.

Separate encoders for document and summary. We conjectured that encoding a document for cross-attention in the word generator, and encoding a summary for generating the next summary sentence representation require extracting different pieces of information. We therefore added a second encoder for the summary generated so far, and initialized it with BERT. This change did not improve over sharing the encoder weights for the article and the summary. However, it introduced many additional parameters, so we discarded this idea.

Same preprocessing for the summary. BERTSUMEXTABS uses different preprocessing formats for the source document and the summary. For the document, every sentence is surrounded by a leading `CLS` token and a trailing `SEP` token. The summary is preceded by a *beginning of summary* token, the summary sentences are separated by a sentence separator token and the end is marked with an *end of summary* token.

We tried homogenizing the preprocessing formats for the document and the summary, such that the encoder does not need to deal with different inputs. We surround every sentence with a `CLS` and `SEP` token. The end of the summary is still marked with an *end of summary* token to tell the decoder to stop.

We did not reach the results of the preprocessing used in BERTSUMEXTABS with this format. Interestingly, the generated summaries consistently contained fewer sentences on average. We conjecture that this could be an artefact of decoding with beam search, but cannot substantiate this presumption.

Contextual sentence representations. In our model, we encode summary sentences individually, without self-attention to the surrounding sentences. It is not possible to allow representations to see future ground-truth sentences, as that would serve as a shortcut for the model and prevent proper learning of the task. While it is possible for the sentence representations to encode information of previous summary sentences, experiments showed no improvements with this change.

Attention to the sentence representation. A different way to integrate the sentence representation in the word generator is to perform attention over it. We experimented with two methods. On the one hand, we specialized an attention head to exclusively look at the sentence representation, while the others attend to the source document. This method performed slightly worse than the base model on ROUGE scores. On the other hand, we concatenated the sentence representation to the encoder outputs, and jointly attended to it in the word generator's cross-attention. When analyzing the attention weights, we realized that the sentence representation was mostly ignored. As a remedy, we separated training into two phases. In the first phase, we trained our model without attention to the document, such that the sentence planner gets a chance to learn meaningful sentence representations and is not ignored from the start. We then finetuned the model with attention to the document. While this increased the attention weights of the sentence representation substantially, the results did not improve over the baseline with the same number of total training steps (pretraining and finetuning combined).

B Preprocessing on Curation Corpus

We follow the instructions in the Curation Corpus Github repository[7] to download the 40000 article-summary pairs. After filtering examples where either the article or the summary are empty, we are left with 39911 examples. We split them into train/validation/test sets as 80/10/10 to arrive at split sizes of 31929/3991/3991.

Since the text extractor from the HTML websites inserts a lot of newlines (probably due to the website layout), we replace them with spaces in order to not split sentences in the middle.

We use the NLTK tokenizer (Bird et al., 2009) to split the article text into sentences. We then preprocess the data in the same way as Liu and Lapata (2019) processed the CNN/DailyMail corpus, except that we do not filter examples based on the number of tokens in the article or summary, but instead keep them irrespective of their length.

We are happy to assist with reconstructing the dataset as we have used it in this paper.

[7] https://github.com/CurationCorp/curation-corpus

Model	ROUGE		
	R-1	R-2	R-L
BSEA (Liu and Lapata, 2019)	42.95 (0.14)	17.67 (0.19)	37.46 (0.21)
BSEA (our implementation)	43.37 (0.37)	17.92 (0.17)	37.73 (0.31)
Sentence planner	**44.40** (0.14)	**18.31** (0.13)	**38.69** (0.10)

Table 8: Comparison of generated with reference summaries on Curation Corpus. Mean and std (in brackets) over 5 runs. Best result in bold.

Model	Sentences		Novel Bigrams	Corefs
	Number	Length		
Gold summaries	*3.46*	*28.0*	*69.22%*	*0.441*
BSEA (Liu and Lapata, 2019)	2.73 (0.09)	27.3 (0.5)	36.77% (0.94%)	0.267 (0.011)
BSEA (our implementation)	2.76 (0.10)	28.5 (0.8)	37.29% (1.32%)	0.283 (0.026)
Sentence planner	3.15 (0.11)	28.2 (0.5)	**39.29%** (2.00%)	**0.289** (0.023)

Table 9: Properties of generated summaries on Curation Corpus. Mean and std (in brackets) over 5 runs.

Dataset / Model	Focus	Coherence
CNN/DailyMail	*0.654*	*0.298*
Curation Corpus	*0.848*	*0.563*
BSEA (Liu and Lapata, 2019)	0.838	0.547
BSEA (our implementation)	0.850	0.563
Sentence planner	0.859	0.562

Table 10: Focus and coherence scores of SUM-QE. Models are trained and evaluated on Curation Corpus. Mean over 5 runs.

C Full Results on Curation Corpus

Tables 8 and 9 show the mean and standard deviation (in brackets) over 5 runs of each model, with random seeds from 1 to 5.

D SUM-QE Evaluation

In line with our evaluations, SUM-QE (Xenouleas et al., 2019) evaluates the linguistic quality of a summary. In particular, the two qualities *focus* and *coherence* are desired properties for natural summaries. However, we found the metric to give non-discriminative scores to all summaries (including reference summaries). We therefore only provide the results for completeness.

SUM-QE automatically evaluates summaries with regard to linguistic quality questions asked in the DUC-05/06/07 tasks. We select the qualities regarding focus and coherence, described as follows:

Q4 – Focus. The summary should have a focus; sentences should only contain information that is related to the rest of the summary.

Q5 – Structure and Coherence. The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic.

The raters were asked to judge summaries on an integer scale of 1 to 5, which is normalized to $(0, 1)$ by the SUM-QE model. It is trained on the raters' judgments and achieves high correlations on a held-out test set. We use the model trained on DUC-05/06 (and evaluated on DUC-07) with the "multi-task-5" setting, producing one output per linguistic quality.

Table 10 holds the SUM-QE scores for the reference summaries of CNN/DailyMail and Curation Corpus. There is an evident difference in scores between the two datasets, with Curation Corpus's summaries being judged more focused and coherent by the model. When comparing the scores of Curation Corpus's reference summaries with the models' scores, there are only minimal differences. The same holds true for a comparison between models. We therefore decided to remove this evaluation from the main text of the paper.

E Example Summaries

Tables 11 and 12 show example summaries from the Curation Corpus validation set for the sentence planner and BERTSUMEXTABS (our implementation), alongside the source article and the reference summary.

Source article Theresa May's plans for a post-Brexit trade deal with the US will be put at risk if she retains EU protections for food and drink such as Champagne and Parma Ham, a senior ally of Donald Trump has warned. The Telegraph has learned that Liam Fox, the International Trade Secretary, has written to David Davis, the Brexit Secretary, warning him not to concede over the issue during negotiations with Brussels. During a recent visit to the US he was told by Paul Ryan, a senior Republican and Speaker of the House of Representatives, that the UK must be able to "diverge" from EU protected status standards to reach a free trade deal. The US produces its own Feta, Parmesan and Champagne and has strongly resisted attempts to ban the sale of American products in the past. Its refusal to compromise on the issue led to the collapse of a major trade deal between the EU and the US. However Michel Barnier, the EU's chief Brexit negotiator, is demanding that Britain must recognise 3,300 protected food and drink products after Brexit. The products are protected under a system of "geographical indications", meaning that they cannot be produced elsewhere.
Reference summary A post-Brexit trade deal with the US may be jeopardised if the UK continues to recognise EU protected status standards for food and drink. The US has resisted calls to adopt protections for products such as feta, Parmesan and Champagne, and would expect the UK to also diverge from them. However, the EU's chief Brexit negotiator, Michel Barnier, says Britain must retain the protections.
Candidate summary (sentence planner) uk prime minister theresa may ' s plans for a post - brexit trade deal with the us will be placed at risk if she retains eu protections for food and drink products such as champagne and parma ham , according to unnamed sources . european trade secretary liam fox has written to david davis , the eu ' s chief brexit negotiator michel barnier , to call for britain to recognise 3 , 300 protected food and drinks products after brexit . the uk produces its own feta , parmesan and champagne imports , and called for the uk to " diverge " from eu protected status standards .
Candidate summary (BERTSUMEXTABS, our implementation) brexit negotiator liam fox has written to david davis , the uk ' s brexit negotiator , calling for britain to recognise 3 , 300 protected food and drink products after brexit . the uk produces its own feta , parmesan and champagne and has strongly opposed attempts to ban the sale of us products in the past . michel barnier , the eu ' s chief brexit negotiator for brexit negotiator michel barnier is calling for the uk to recognise three , 300 products following brexit .

Table 11: Hard example from the Curation Corpus. The sentence planner correctly calls "a senior ally of Donald Trump" an "unnamed source". It nicely includes the Speaker of the House's demand to "diverge" from EU standards as a call by the US. It gets confused with the International Trade Secretary, the Brexit Secretary and the EU's chief Brexit negotiator. It also mistakes the US for the UK when talking about a country producing its own products. BERTSUMEXTABS does these same mistakes, but gets even more confused with the Brexit negotiator. It repeats the call to recognize the protected products by the Brexit negotiator, and misses the main point of the article, namely that this issue jeopardizes the post-Brexit trade deal.

Source article

FILE PHOTO: U.S. President Donald Trump talks to reporters as he heads to the Marine One helicopter to depart the White House for travel to Florida in Washington, U.S. October 8, 2018. REUTERS/Jonathan Ernst/File PhotoWASHINGTON (Reuters) - President Donald Trump's administration on Thursday for a second time asked the U.S. Supreme Court to put the brakes on a lawsuit filed by young activists who have accused the U.S. government of ignoring the perils of climate change. In the lawsuit, 21 activists, ages 11 to 22, said federal officials violated their rights to due process under the U.S. Constitution by failing to adequately address carbon pollution such as emissions from the burning of fossil fuels. The lawsuit was filed in 2015 against former President Barack Obama and government agencies in a federal court in Eugene, Oregon. Both the Obama and Trump administration have failed in efforts to have the lawsuit thrown out. On July 30, the high court rejected an earlier application by the Trump administration, saying it was premature. Eugene, Oregon-based federal judge Ann Aiken on Monday issued another ruling allowing the case to move forward to trial on Oct. 29 if either the high court or the San Francisco-based 9th U.S. Circuit Court of Appeals do not intervene. Reporting by Lawrence Hurley; Editing by Will Dunham

Reference summary

President Trump's administration on 18 October failed in its attempt to halt a climate change lawsuit filed in 2015 by 21 activists, aged between 11 and 22, which accuses the US government of ignoring the risks of climate change. The Obama administration had also attempted unsuccessfully to have the lawsuit thrown out, while the high court rejected a previous application by the Trump administration to have the case dismissed on 30 July. The latest ruling from the Supreme Court means the case can proceed to trial on 29 October if neither the high court or Court of Appeals intervenes.

Candidate summary (sentence planner)

president donald trump ' s administration has asked the us supreme court to put the brakes on a lawsuit filed by young activists who have accused the us government of ignoring the perils of climate change . in 2015 , 21 activists claimed federal officials violated their rights to due process under the constitution by failing to address carbon pollution , including emissions from the burning of fossil fuels . on 30 july , the high court rejected an earlier application by the trump administration , saying it was premature .

Candidate summary (BERTSUMEXTABS, our implementation)

president donald trump ' s administration for a second time has asked the us supreme court to put the brakes on a lawsuit filed by 21 activists who have accused the us government of ignoring the perils of climate change . the case , which was filed in 2015 against former president barack obama and government agencies in a federal court in oregon , is being brought forward by the high court on 30 july .

Table 12: Example from the Curation Corpus. The sentence planner manages to get all facts correct, and summarizes the important content very well by removing phrases such as "on Thursday for a second time", "U.S." in "U.S. Constitution" and "adequately" in "adequately address". It also uses the information that the lawsuit was filed in 2015 from a later sentence to include in the sentence about the origin of the lawsuit. BERTSUMEXTABS also nicely fuses information in its first generated sentence. In the second one, however, it mistakenly believes that the case will be handled on July 30, instead of October 29. It is again a bit shorter on information compared to the sentence planner.